ELSEVIER
Theoretical Computer Science 194 (1998) 35–55
Theoretical Computer Science

Remarks on regulated limited ET0L systems and regulated context-free grammars

Henning Fernau [a,1], Dietmar Wätjen [b,*]

[a] *Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Sand 13, D-72076 Tübingen, Germany*

[b] *Institut für Theoretische Informatik, Technische Universität Braunschweig, Fallersleber-Tor-Wall 22, Postfach 3329, D-38023 Braunschweig, Germany*

Received April 1996; revised October 1996
Communicated by A. Salomaa

Abstract

We continue the studies of the second author on regulated uniformly k-limited and regulated k-limited ET0L systems. We focus on the permitting and forbidding random context regulation. Especially, we establish some results on (regulated) propagating (uniformly) k-limited ET0L systems which were not solved in [2, 18, 21, 22]. Moreover, relations to recurrent programmed languages introduced by von Solms are exhibited.

Keywords: Formal languages; Limited T0L systems; Regulated rewriting; Regulated context-free grammars

1. Introduction

The regular rewriting of language-generating devices has been extensively studied in the literature. There have been considered, among others, programmed, matrix, periodically time-varying, regular controlled or random context grammars, L systems or different kinds of limited L systems. A comprehensive representation of regulated grammars and L systems can be found in the monograph of Dassow and Păun [3], regulated k-limited ET0L systems are considered in [18, 20], regulated uniformly-k-limited ET0L systems in [19, 21]. In case of grammars, ET0L systems or limited ET0L systems, it has been shown that with sufficiently large regulations, all these devices generate the family of recursively enumerable languages.

In this paper, we shall give some further results concerning regulated (uniformly) k-limited ET0L systems. Especially, we consider uniformly k-limited ET0L systems with forbidding random context which are only mentioned implicitly in [21]. In case

* Corresponding author.

[1] Supported by Deutsche Forschungsgemeinschaft grant DFG La 618/3-1.

PII S0304-3975(96)00329-5

that the limitation k is greater than 1, forbidding random context is already as powerful as random context with appearance checking. Furthermore, we exhibit some relations to recurrent programmed grammars as defined by von Solms in [15] in case of (uniformly) k-limited ET0L systems with permitting random context. To this end, we establish some new properties of recurrent context-free programmed languages. There are interesting links with one of the old open questions of formal language theory, namely whether permitting random context grammars are as powerful as programmed grammars without appearance checking; more precisely, we show a number of (possibly) intermediate classes, including recurrent context-free programmed languages without appearance checking.

In the sequel, we denote by $\mathbb{N}$ the set of all natural numbers (where $0 \notin \mathbb{N}$). Then $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

2. Definitions

Uniformly k-limited T0L or ET0L systems are considered in [17, 22], while k-limited ET0L systems have been introduced in [16]. Both kinds of systems present limitations of the parallel rewriting in 0L systems. For the convenience of the reader, we repeat their definitions. A *uniformly k-limited* ET0L *system* (abbreviated as uklET0L *system*) $G=(\Sigma, H, \omega, \Delta, k)$ is given by $k \in \mathbb{N}$ and an ET0L system $(\Sigma, H, \omega, \Delta)$ with *alphabet* Σ, finite set of *tables* H (where a table is a finite substitution on Σ), *axiom* $\omega \in \Sigma^*$, and *terminal alphabet* $\Delta \subseteq \Sigma$. For $w, v \in \Sigma^*$, a derivation step $w \Rightarrow v$ according to the uklET0L system G is given by a step $w \Rightarrow_h v$ for some $h \in H$ where v arises from w by substituting exactly $\min\{k, |w|\}$ occurrences of symbols in the word w according to h where $|w|$ is the length of w. Let $\Rightarrow^*$ be the reflexive transitive closure of $\Rightarrow$. Then $L(G)=\{w \in \Delta^* \mid \omega \Rightarrow^* w\}$ is the uklET0L *language generated by G*. A k-limited ET0L system (abbreviated as klET0L *system*) is defined analogously with the exception that at each step of the rewriting process, exactly $\min\{k, \#_a w\}$ occurrences of each symbol a in the word w considered have to be rewritten, where $\#_a w$ is the number of occurrences of the symbol a in w. By $\mathscr{L}$(uklET0L) we denote the *family of all* uklET0L *languages*, by $\mathscr{L}$(klET0L) the *family of all* klET0L *languages*. As usual, we also consider propagating such systems. The corresponding language families are denoted by $\mathscr{L}$(uklEPT0L) or $\mathscr{L}$(klEPT0L), respectively. Sometimes, we use parentheses notations like $\mathscr{L}(1\text{lE(P)T0L}) = \mathscr{L}(\text{P}, ut, \text{cf}(-\varepsilon))$ in order to say that the equation holds both in the case of excluding erasing productions (as indicated by the P and $-\varepsilon$ enclosed in parentheses) and in the case of admitting erasing productions (neglecting the parenthesis contents).

The definitions of matrix (m), periodically time-varying (ptv), programmed (p), graph-controlled (gc) and regularly controlled (rc) uklET0L systems are given in [19] and [21] and shall not be repeated here. The corresponding definitions of regulated klET0L systems can be found in [18]. For the case of uklET0L systems we recall the definition of random context systems. We assume that to the set of tables H of a uklET0L system there is associated a finite set of labels Lab(H) such

that every label belongs to exactly one table and every table possesses at least one label. Sometimes we do not distinguish between labels and the corresponding tables. $G=(\Sigma,H,\omega,\Delta,k,oc,noc)$ is a *random context* uklET0L *system with appearance checking* if $(\Sigma,H,\omega,\Delta,k)$ is a uklET0L system and $oc:\mathrm{Lab}(H)\to\mathfrak{P}(\Sigma)$ and $noc:\mathrm{Lab}(H)\to\mathfrak{P}(\Sigma)$ are mappings ($\mathfrak{P}(\Sigma)$ being the power set of Σ). A word $w\in\Delta^*$ is derived according to G if there exists a derivation

$$D:\omega=w_0\Rightarrow_{h_{i_1}}w_1\Rightarrow_{h_{i_2}}\cdots\Rightarrow_{h_{i_n}}w_n=w$$

with $h_{i_v}\in H$, $v=1,\ldots,n$, for some $n\in\mathbb{N}_0$ such that all letters of $oc(h_{i_{v+1}})$ (the *occurrence set* of $h_{i_{v+1}}$) occur in w_v and no letter of $noc(h_{i_{v+1}})$ (the *non-occurrence set* of $h_{i_{v+1}}$) occurs in w_v, $v=0,1,\ldots,n-1$. Because we have a uniform limitation, it might be possible that a derivation step is executed without substituting any occurrence of a fixed symbol of the occurrence set of the table used in this step. Let $\mathscr{L}(rand\text{-}app,$ uklET0L) be the corresponding language family. If $noc(h)=\emptyset$ for all $h\in\mathrm{Lab}(H)$, then the appearance checking is deleted, we speak of *permitting random context* and we write $\mathscr{L}(rand,$ uklET0L). If $oc(h)=\emptyset$ for all $h\in\mathrm{Lab}(H)$, we speak of *forbidding random context*, and we write $\mathscr{L}(frand,$ uklET0L).

Obviously, we may consider the random context systems in combination with the other regulated systems. As an example, we mention a *random context graph-controlled* uklET0L *system with appearance checking*. A derivation of such a system has to fulfill the properties of both systems. More generally, for $x\in\{\emptyset,p,ptv,m,gc,rc\}$, $y\in\{\emptyset,rand,frand,rand\text{-}app\}$, the corresponding language families are denoted by $\mathscr{L}(x,y,$ uklET0L) where in case $x=\emptyset$ or $y=\emptyset$ the corresponding letter $\emptyset$ should be omitted. Accordingly, the different regulation mechanisms are defined for klET0L systems. We use these regulation mechanisms also in connection with context-free grammars. The control is imposed upon the productions of the grammars, tables do not exist. Neglecting this difference, the definitions can be directly carried over. We write $\mathscr{L}(x,y,\mathrm{cf})$ for $x\in\{\emptyset,p,ptv,m,gc,rc\}$ and $y\in\{\emptyset,rand,frand,rand\text{-}app\}$. We note that the regulation mechanisms defined in this manner do not always coincide with those given in [14] or [3].

Indeed, in the sequel we use the definition of a programmed grammar as in [3]. Thus, a *context-free programmed grammar* $G=(V_N,V_T,X_0,P,\sigma,\mu)$ *with appearance checking* is given by a context-free grammar (V_N,V_T,X_0,P) and mappings $\sigma,\mu:\mathrm{Lab}(P)\to\mathfrak{P}(\mathrm{Lab}(P))$. We say that (w_1,f_1) directly derives (w_2,f_2), $w_1\in(V_N\cup V_T)^+$, $w_2\in(V_N\cup V_T)^*$, $f_1,f_2\in\mathrm{Lab}(P)$ (written $(w_1,f_1)\to(w_2,f_2)$), if either the context-free production labelled by f_1 is not applicable to w_1 and $w_1=w_2$, $f_2\in\mu(f_1)$, or else $w_1\Rightarrow_{f_1}w_2$ (This means that the rule labelled with f_1 is actually applied to w_1, yielding w_2.) and $f_2\in\sigma(f_1)$. The language generated by G consists of all words $u\in V_T^*$ such that there is a derivation

$$(X_0,f_0)\Rightarrow(w_1,f_1)\Rightarrow\cdots\Rightarrow(w_n,f_n)=(u,f_n)$$

for some $n\in\mathbb{N}$, $f_0\in\mathrm{Lab}(P)$. A production $f:A\to w$ together with its success field $\sigma(f)$ and failure field $\mu(f)$ is also written as $f=(A\to w,\sigma(f),\mu(f))$. By $\mathscr{L}(\mathrm{P},ac,\mathrm{cf})$

we denote the corresponding language family. If G is ε-free we write $\mathscr{L}(\mathrm{P}, ac, \text{cf-}\varepsilon)$. If $\mu(f)=\emptyset$ for all $f \in \mathrm{Lab}(P)$, the appearance checking is deleted, and the letters ac are deleted in the notations. In this case, a corresponding grammar can be also denoted by $G=(V_N, V_T, X_0, P, \sigma)$. If $\sigma(f)=\mu(f)$ for all $f \in \mathrm{Lab}(P)$, the grammar is said to be with unconditional transfer, and the letters ut substitute ac in our notations.

Especially, von Solms [15] considered recurrent context-free programmed grammars. A context-free programmed grammar G is a *recurrent context-free programmed grammar* if for every $f=(A \to w, \sigma(f), \mu(f))$ of G, if $\mu(f)=\emptyset$, then $f \in \sigma(f)$, and if $\mu(f) \neq \emptyset$, then $f \in \sigma(f)=\mu(f)$. The corresponding language families are denoted by $\mathscr{L}(\mathrm{RP}, ac, \mathrm{cf}(\text{-}\varepsilon))$ where the appearance checking may be deleted.

3. Preliminary results on regulated rewriting

Observe that our definition of a random context context-free grammar differs from that used in [3] and in many other papers (our definition is predominant in the Russian literature where the concept of random context was extensively studied). In [3], if a production $X \to w$ has to be applied to a word PXQ, then the occurrence and non-occurrence sets of this production refer to PQ. The corresponding language family is written as $\mathscr{L}(\mathrm{RC}, ac, \mathrm{cf})$. If the occurrence or non-occurrence sets are empty, we write $\mathscr{L}(\mathrm{fRC}, \mathrm{cf})$ or $\mathscr{L}(\mathrm{RC}, \mathrm{cf})$, respectively. If all grammars are supposed to be ε-free, we use the notation cf-ε. Both forms of random context context-free grammars are equivalent:

Theorem 3.1. *We have*

$$\mathscr{L}(\textit{rand-app}, \mathrm{cf}(\text{-}\varepsilon)) = \mathscr{L}(\mathrm{RC}, ac, \mathrm{cf}(\text{-}\varepsilon)), \quad \mathscr{L}(\textit{rand}, \mathrm{cf}(\text{-}\varepsilon)) = \mathscr{L}(\mathrm{RC}, \mathrm{cf}(\text{-}\varepsilon)),$$
$$\textit{and } \mathscr{L}(\textit{frand}, \mathrm{cf}(\text{-}\varepsilon)) = \mathscr{L}(\mathrm{fRC}, \mathrm{cf}(\text{-}\varepsilon)).$$

Proof. Let $G=(V_N, V_T, X_0, P, oc, noc)$ be a random context context-free grammar with appearance checking, in the sense of our paper. Every production $p: X \to w$ of P can be only applied if X occurs in the word considered. Thus, by deleting every production $p: X \to w$ with $X \in noc(p)$ from P and by defining $oc'(p')=oc(p') \setminus \{X'\}$ for every other production $p': X' \to w'$ (we collect all these productions p' into P'), we get an equivalent grammar $G'=(V_N, V_T, X_0, P', oc', noc')$ which is also a random context context-free grammar in the sense of [3]. If $oc(p)=\emptyset$ or $noc(p)=\emptyset$ for all $p \in P$, then $oc'(p')=\emptyset$ or $noc'(p')=\emptyset$, respectively, for all $p' \in P'$.

For the other direction, let $G=(V_N, V_T, X_0, P, oc, noc)$ be a random context context-free grammar in the sense of [3]. Every production $p: X \to w$ with $X \in oc(p)$ is replaced by the productions $p_1: X \to X_p$ with a new symbol X_p where $oc'(p_1)=\emptyset$, $noc'(p_1)=\{X_{p'} \mid p' \in P\}$, and $p_2: X_p \to w$ with $oc'(p_2)=oc(p)$, $noc'(p_2)=noc(p)$. If $noc(p')=\emptyset$ for all $p' \in P$, then we set $noc'(p_1)=\emptyset$ instead. Other productions (i.e.,

productions $p:X \to w$ with $X \notin oc(p)$) and their occurrence and non-occurrence sets remain unchanged with the exception of the case $noc(p) \neq \emptyset$, where we set

$$noc'(p)=noc(p) \cup \{X_{p'} \mid X \in noc(p), p' \in P\}.$$

Obviously, we get an equivalent random context context-free grammar (with appearance checking) in the sense of this paper.

If $oc(p)=\emptyset$ for all $p \in P$, then $oc'(p')=\emptyset$ for all $p' \in P'$. If $noc(p)=\emptyset$ for all $p \in P$, then we set $noc'(p_1)=\emptyset$ in our construction above, such that $noc'(p')=\emptyset$ for all $p' \in P'$.

The arguments are also true in the ε-free case. □

In the following, we show that every permitting random context context-free grammar can be simulated by a recurrent programmed grammar without appearance checks.

Theorem 3.2. *We have* $\mathscr{L}(rand, \mathrm{cf}(\text{-}\varepsilon)) \subseteq \mathscr{L}(\mathrm{RP}, \mathrm{cf}(\text{-}\varepsilon))$.

Proof. Let $G=(V_N, V_T, X_0, P, oc)$ be a random context context-free grammar. We define a recurrent programmed grammar $G'=(V_N', V_T, X_0, P', \sigma)$ with $V_N'=V_N \cup \{F\}$. Consider a production $r: A \to w$ of G with $oc(r)=\{A_1, \ldots, A_n\}$ for some $n \in \mathbb{N}_0$. Every such production r is simulated by the following set of productions:

$$p_{r,i}=(A_i \to A_i, \{p_{r,i}, p_{r,i+1}\}) \quad \text{for } 1 \leqslant i \leqslant n,$$

$$p_{r,n+1}=(A \to w, \{p_{r,n+1}\} \cup \{p_{\bar{r},1} \mid \bar{r} \in \mathrm{Lab}(P)\}).$$

Obviously, $L(G)=L(G')$. □

Unfortunately, we do not know whether a similar statement is valid if we further allow appearance checks.

However, our last result is interesting on its own right, since it connects one of the classical open questions in formal language theory, namely whether permitting random context grammars are as powerful as programmed grammars without appearance checks or not, since now there is an intermediate class, namely $\mathscr{L}(\mathrm{RP}, \mathrm{cf}(\text{-}\varepsilon))$.

Theorem 3.3. *We have the strict inclusion*

$$\mathscr{L}(\mathrm{RP}, \mathrm{cf}(\text{-}\varepsilon)) \subsetneq \mathscr{L}(\mathrm{RP}, ac, \mathrm{cf}(\text{-}\varepsilon)).$$

Proof. In Lemmas 5.3 and 5.4 of [15], it is proved that $\mathscr{L}(rand, \mathrm{EPT0L})=\mathscr{L}(\mathrm{RP}, ac, \mathrm{cf}\text{-}\varepsilon)$. It is easily seen that the corresponding proof is also valid in the non-propagating case. Obviously,

$$L=\{a^{2^n} \mid n \in \mathbb{N}_0\} \in \mathscr{L}(\mathrm{EPT0L}) \subseteq \mathscr{L}(rand, \mathrm{EPT0L}).$$

It has been proved (see [12, p. 727]) that $L \notin \mathscr{L}(\mathrm{M}, \mathrm{cf})$ where $\mathscr{L}(\mathrm{M}, \mathrm{cf})$ is the family of context-free matrix languages. Since $\mathscr{L}(\mathrm{M}, \mathrm{cf}(\text{-}\varepsilon))=\mathscr{L}(\mathrm{P}, \mathrm{cf}(\text{-}\varepsilon))$ (see [3, Theorem 1.2.2]), it follows that $L \notin \mathscr{L}(\mathrm{RP}, \mathrm{cf}(\text{-}\varepsilon))$. □

In the following, we want to establish some new closure properties of recurrent programmed languages. To this end, we need the following notion.

A *one-input finite state transducer with accepting state*, or *1-a-transducer* for short, is a 6-tuple $M=(Q,X,Y,\delta,q_0,Q_f)$, where Q is a finite set of states, X and Y are finite (input and output) alphabets, $q_0\in Q$ is the initial state, $Q_f\subseteq Q$ is the set of accepting or final states, and δ is a finite subset of $Q\times(X\cup\{\varepsilon\})\times(Y\cup\{\varepsilon\})\times Q$. M is called *ε-free* if $\delta\subseteq Q\times(X\cup\{\varepsilon\})\times Y\times Q$.

By a *computation* of such a 1-a-transducer a word $h=h_1\cdots h_n\in\delta^+$ is understood such that

1. $\mathrm{pr}_1(h_1)=q_0$, $\mathrm{pr}_4(h_n)\in Q_f$ and
2. $\mathrm{pr}_1(h_{i+1})=\mathrm{pr}_4(h_i)$ for all i, $i=1,\ldots,n-1$,

where pr_i are projection homomorphisms on δ^* defined by

$$\mathrm{pr}_i((x_1,x_2,x_3,x_4))=x_i \quad \text{for } i=1,2,3,4.$$

The set of all computations of M is denoted by $C(M)$. A *1-a-transducer mapping* is defined for each language $L\subseteq X^*$ by $M(L)=\mathrm{pr}_3(\mathrm{pr}_2^{-1}(L)\cap C(L))$.

Remark. By [11, Theorem 3.2.1], a family of languages $\mathscr{L}$ is a trio if and only if it is closed under ε-free 1-a-transducer mappings. $\mathscr{L}$ is a full trio if and only if it is closed under 1-a-transducer mappings.

Theorem 3.4. *$\mathscr{L}$(RP, cf) is a full trio. $\mathscr{L}$(RP, cf-ε) is a trio.*

Proof. Let $G=(V_N,V_T,S,P,\sigma)$ be some recurrent context-free programmed grammar without appearance checking. Let $M=(Q,V_T,V_T',\delta,q_0,Q_f)$ be some 1-a-transducer. We construct a recurrent context-free programmed grammar without appearance checking

$$G'=(V_N',V_T',S',P',\sigma') \quad \text{with } L(G')=M(L(G)).$$

The proof idea mainly follows the classical triple construction proving the closure of certain basically context-free language classes under intersection with regular sets. Let

$$V_N'=\{S'\}\cup(Q\times(V_N\cup V_T\cup\{\varepsilon\})\times Q).$$

The new set of production labels consists of

$$\begin{aligned}\mathrm{Lab}(P') =\ & Q_f\cup\Lambda\cup\delta\\ & \cup\{(p,q_0,\ldots,q_n)\mid p\in\mathrm{Lab}(P);(p:A\to w,\sigma)\in P;\\ & \quad |w|=n>0;q_0,\ldots,q_n\in Q\}\\ & \cup\{(p,q_0,q_1)\mid p\in\mathrm{Lab}(P);(p:A\to\varepsilon,\sigma)\in P;q_0,q_1\in Q\},\end{aligned}$$

where

$$\Lambda=(Q\times(V_T\cup\{\varepsilon\})\times Q\times Q)\cup(Q\times Q\times(V_T\cup\{\varepsilon\})\times Q).$$

More precisely, for every $q_f \in Q_f$, we have a special start production

$$(q_f : S' \to (q_0, S, q_f), \mathrm{Lab}(P')),$$

and, for every $(q_1, x_1, x_2, q_2) \in \delta$, terminating productions

$$((q_1, x_1, x_2, q_2) : (q_1, x_1, q_2) \to x_2, \delta).$$

(Here, we assume without loss of generality that $\delta \supseteq \{(q, \varepsilon, \varepsilon, q) \mid q \in Q\}$, if M is not ε-free. This is recognized as follows. In general, G is not ε-free. Then it is possible that we introduce non-terminals of the form (q, ε, q) by applying a production defined in Eq. (1) below. Therefore, the non-ε-free transducer is given an empty input when it is in state q. Now, there are two correct possibilities: either the transducer makes a "real" ε-move according to its transition relation δ, or, it simply ignores the empty word in the input and keeps staying in state q. The latter case is included in the first one if we presume $(q, \varepsilon, \varepsilon, q) \in \delta$.)

Each production $(p : A \to w, \sigma) \in P$ with $w \neq \varepsilon$, i.e., $w = w_1 \dots w_n$ such that $w_i \in V_N \cup V_T$ for $1 \leqslant i \leqslant n$ is simulated by one of the following productions:

$$((p, q_0, \dots, q_n) : (q_0, A, q_n) \to (q_0, w_1, q_1)(q_1, w_2, q_2) \dots (q_{n-1}, w_n, q_n), \sigma'),$$

where

$$\begin{aligned} \sigma' = {} & \{(p', q'_0, \dots, q'_m) \mid p' \in \sigma; (p' : A' \to w', \sigma(p')) \in P; \\ & |w'| = m > 0; q'_0, \dots, q'_m \in Q\} \\ & \cup \{(p', q'_0, q'_1) \mid p' \in \sigma; (p' : A' \to \varepsilon, \sigma(p')) \in P; q'_0, q'_1 \in Q\} \cup \Lambda \cup \delta, \end{aligned}$$

and $q_0, \dots, q_n \in Q$.

Each production $(p : A \to \varepsilon, \sigma) \in P$ is simulated by one of the following productions:

$$((p, q_0, q_1) : (q_0, A, q_1) \to (q_0, \varepsilon, q_1), \sigma'), \tag{1}$$

where σ' is defined as above, and $q_0, q_1 \in Q$.

Moreover, we have a number of productions dealing with the possible ε-moves of the transducer M. For each $q, q', q'' \in Q$ and $B \in V_T \cup \{\varepsilon\}$, we have

$$((q, B, q', q'') : (q, B, q'') \to (q, B, q')(q', \varepsilon, q''), \Lambda \cup \delta),$$

and

$$((q, B, q', q'') : (q, B, q'') \to (q, \varepsilon, q')(q', B, q''), \Lambda \cup \delta).$$

In this way, it is possible to simulate ε-moves "to the right" and "to the left" of some nondeterministically guessed state-coloured terminal symbol. □

In the following sections, we show the interrelations between uklET0L and klET0L languages (with some random context conditions) and appropriate classes defined via programmed grammars.

4. *k*lET0L and u*k*lET0L (without random context)

Most of the contents of this section is already known; we include these results mainly for reasons of completeness.

Theorem 4.1. *We have* $\mathscr{L}(1\text{lE(P)T0L}) = \mathscr{L}(\text{P}, ut, \text{cf}(\text{-}\varepsilon))$.

The proof below is also contained in the conference paper [9, Theorem 3.3 (2)].

Proof. All but the inclusion $\mathscr{L}(1\text{lEPT0L}) \supseteq \mathscr{L}(\text{P}, ut, \text{cf-}\varepsilon)$ has already been shown in [2, Theorem 1]. Below, we show the missing relation. Let $G=(V_N, V_T, S, P, \sigma, \sigma)$ be a programmed grammar with unconditional transfer without erasing productions. Let $P=\{(p_j : A_j \to w_j, \sigma_j, \sigma_j) \mid 1 \leqslant j \leqslant m\}$ be the set of labelled productions of G. The set of labels (nodes) is denoted by Lab(P). Let $V_G = V_N \cup V_T$. We construct a 1lEPT0L system $G'=(\Sigma, H, S', V_T, 1)$ simulating G as follows. Let

$$\Sigma = V_G \cup \{S', F\} \cup \{[a, p] \mid a \in V_G, p \in \text{Lab}(P)\}$$
$$\cup \{a', a'', a''' \mid a \in V_G\} \cup \{\tilde{A}, \hat{A}, \bar{A} \mid A \in V_N\},$$
$$H = \{h_I, h_T\} \cup \{h_p, h_{p,1}, \ldots, h_{p,4} \mid p \in \text{Lab}(P)\}.$$

The tables are defined as follows. (We include only such productions which do not lead to the failure symbol F; they have to be supplemented otherwise because of the completeness condition inherited from L systems.)

The initialization table h_I embraces

$$h_I(S') = \{v[a, p] \mid (S, q) \overset{*}{\Rightarrow}_G (va, p) \text{ and } a \in V_G, |v| \geqslant 2; q, p \in \text{Lab}(P)\}$$
$$\cup \{w \mid (S, q) \overset{*}{\Rightarrow}_G (w, p) \text{ and } w \in V_T^2 \cup V_T; q, p \in \text{Lab}(P)\}.$$

The termination table h_T contains $h_T([x, p]) = x$ for $x \in V_T$, $p \in \text{Lab}(P)$. Note that a premature attempt to apply the termination table to a string still containing "real" nonterminal symbols would inevitably introduce the failure symbol F.

During a simulation, we face a string of the form $w_1[x, p_j]w_2$. This testifies that in G there is a derivation $(S, q) \overset{*}{\Rightarrow} (w_1 x w_2, p_j)$. Since the other case has already been treated in the initialization table, we assume $|w_1 w_2| \geqslant 2$.

The first and simple case we deal with is $A_j \neq x$. Now, the marker can stay at the symbol x in its place, and the actual simulation takes place elsewhere. This is accomplished with the table h_{p_j} which contains

$$h_{p_j}([x, p_j]) = \{[x, q] \mid q \in \sigma_j\} \quad \text{for } x \in V_G \setminus \{A_j\},$$
$$h_{p_j}(A_j) = \{w_j\},$$
$$h_{p_j}(Y) = Y \quad \text{for } Y \in V_G \setminus \{A_j\}.$$

Observe that the "unconditional transfer" is done automatically via the definition of a derivation step in 1lEPT0L systems.

The second case, $A_j = x$, is more complicated. Why? It is possible that there is another A_j in the string $w_1 w_2$ not hidden in the disguise $[A_j, p_j]$. Both forms of A_j should have a chance to be chosen to take part in the rewriting $A_j \to w_j$ which has to be simulated.

This is accomplished by the following four tables.

1. $h_{p_j,1}$ contains $h_{p_j,1}([A_j, p_j]) = \{\hat{A}_j\}$, $h_{p_j,1}(A_j) = \{\tilde{A}_j\}$, and we have for each $Y \in V_G \backslash \{A_j\}$, marking productions $h_{p_j,1}(Y) = \{Y'\}$.

2. $h_{p_j,2}$ contains $h_{p_j,2}(\hat{A}_j) = \{\bar{A}_j\}$, $h_{p_j,2}(\tilde{A}_j) = \{\bar{A}_j\}$, $h_{p_j,2}(A_j) = \{A_j''\}$ and, for every $Y \in V_G \backslash \{A_j\}$, marking productions $h_{p_j,2}(Y') = \{Y''\}$, and $h_{p_j,2}(Y) = \{Y\}$ [in order to conserve unmarked symbols]. Note that after a successful application of this table, at least one and at most two occurrences of $\bar{A}_j$ are present. Moreover, there is at least one occurrence of the form a'' for some $a \in V_G$, since we presume $|w_1[x, p_j]w_2| \geqslant 3$, and we have monotone productions only.

3. $h_{p_j,3}$ contains $h_{p_j,3}(A_j') = \{'_j\}$, and, for every $Y \in V_G$, $h_{p_j,3}(Y) = \{Y\}$ and $h_{p_j,3}(Y'') = \{Y'''\}$.

4. $h_{p_j,4}$ embraces $h_{p_j,4}(A_j') = \{v[a,q] \mid w_j = va, a \in V_G, q \in \sigma_j\}$, $h_{p_j,4}(\bar{A}_j) = \{A_j\}$, and $h_{p_j,4}(Y''') = h_{p_j,4}(Y) = \{Y\}$ for $Y \in V_G$. □

Unfortunately, we do not know whether an analogue to Theorem 4.1 is also true for klE(P)T0L languages in general, where k is some arbitrary fixed number. Somehow, 1lE(P)T0L systems seem to be stronger than, say, 2lE(P)T0L systems, but we could not fix this exactly. A similar situation is found when we consider klE(P)T0L systems with permitting random context, see below.

In the case of uklET0L systems, the situation is even worse, since only an inclusion relation is known, see [10, p. 57f]. This is also correct in our case. The next theorem also follows from Theorem 6.4 below.

Theorem 4.2. *We have* $\mathscr{L}(\text{u}k\text{lE(P)T0L}) \subseteq \mathscr{L}(\text{P}, \text{cf}(\text{-}\varepsilon))$.

On the other hand, for some modified notion of uniform limited systems, a characterization of programmed grammars without appearance checks can be obtained [8].

We know that

$$L = \{a^{2^n} \mid n \in \mathbb{N}_0\} \in \mathscr{L}(\text{EPT0L}) \subseteq \mathscr{L}(k\text{lE(P)T0L}) \tag{2}$$

for all $k \in \mathbb{N}$, but on the other hand (see proof of Theorem 3.3), we have $L \notin \mathscr{L}(\text{P}, \text{cf}(\text{-}\varepsilon))$. Thus $L \notin \mathscr{L}(\text{u}k\text{lE(P)T0L})$. It follows that $\mathscr{L}(k\text{lE(P)T0L})$ is not contained in $\mathscr{L}(\text{u}k'\text{lE(P)T0L})$ for all $k, k' \in \mathbb{N}$. In this way, we partially solve a problem marked as open in [22].

Now, we turn our attention towards (uniformly) limited systems regulated by random context conditions.

5. Forbidding random context

In the literature about regulated klET0L and uklET0L systems, forbidding context systems have not been explicitly considered, although in [21], they are present implicitly.

First we turn to the case of k-limited ET0L systems, which is the easy case here. In [18, Theorems 5.2 and 5.4], it is shown that for all $k \in \mathbb{N}$ and all $x \in \{p, ptv, m, gc\}$,

$$\mathscr{L}(rand, k\text{lET0L}) = \mathscr{L}(x, rand, k\text{lET0L}) = \mathscr{L}(x, rand\text{-}app, k\text{lET0L}) = \mathscr{L}(\text{re}). \quad (3)$$

The appearance checking, that is the forbidding random context, is not necessary to get the family of recursively enumerable languages. If no occurrence sets are present, the forbidding random context does not enlarge the generated language family, either.

Theorem 5.1. *For all* $k \in \mathbb{N}$, $\mathscr{L}(frand, k\text{lE(P)T0L}) = \mathscr{L}(k\text{lE(P)T0L})$.

Proof. It is only necessary to prove $\mathscr{L}(frand, k\text{lE(P)T0L}) \subseteq \mathscr{L}(k\text{lE(P)T0L})$. If $G = (\Sigma, H, \omega, \Delta, k, noc)$ is a klE(P)T0L system with forbidding random context, then $G' = (\Sigma \cup \{F\}, H', \omega, \Delta, k)$ is a simulating klE(P)T0L system where $H' = \{h' \mid h \in H\}$ with

$$h'(x) = \{F\} \quad \text{for } x \in noc(h) \cup \{F\} \quad \text{and} \quad h'(x) = h(x) \text{ else.} \qquad \square$$

Now, we turn to uniform limitations. Theorem 4.3 of [21] shows that for $x \in \{p, ptv, m, gc, rc\}$,

$$\mathscr{L}(x, rand\text{-}app, \text{u1lE(P)T0L}) = \mathscr{L}(x, frand, \text{u1lE(P)T0L}). \quad (4)$$

Studying the proof of Theorem 5.2 of [21] we recognize that we can carry it over to the case of forbidding random context. We get

Theorem 5.2. *We have*

$$\mathscr{L}(frand, \text{u1lE(P)T0L}) = \mathscr{L}(frand, \text{cf}(\text{-}\varepsilon)),$$

and furthermore, for all $x \in \{p, ptv, m, gc, rc\}$,

$$\mathscr{L}(x, frand, \text{u1lE(P)T0L}) = \mathscr{L}(x, frand, \text{cf}(\text{-}\varepsilon)) = \mathscr{L}(rc, frand, \text{u1lE(P)T0L}).$$

It is known (cf. [3]) that $\mathscr{L}(\text{P}, ac, \text{cf})$ the family of languages generated by context-free programmed grammars with appearance checking, coincides with the family $\mathscr{L}(\text{re})$ of recursively enumerable languages. $\mathscr{L}(\text{P}, ac, \text{cf-}\varepsilon)$ is known to be strictly included in the family of context-sensitive languages. From Theorem 5.2 together with Eq. (4) and Theorem 5.4 of [21], we derive

Theorem 5.3. *For all* $x \in \{p, ptv, m, gc, rc\}$,

$$\begin{aligned}\mathscr{L}(rand\text{-}app, \mathrm{u1lE(P)T0L}) &= \mathscr{L}(x, rand\text{-}app, \mathrm{u1lE(P)T0L}) \\ &= \mathscr{L}(x, frand, \mathrm{u1lE(P)T0L}) \\ &= \mathscr{L}(x, frand, \mathrm{cf}(\text{-}\varepsilon)) = \mathscr{L}(\mathrm{P}, ac, \mathrm{cf}(\text{-}\varepsilon)).\end{aligned}$$

We see that forbidding random context for u1lET0L systems or context-free grammars leads with at least one further regulation mechanism to the family $\mathscr{L}(\mathrm{re})$ (in the non-propagating case).

For the proof of the next theorem, we need the following lemma.

Lemma 5.4. *For every* $k \in \mathbb{N}$, $\mathscr{L}(frand, \mathrm{u}k\mathrm{lE(P)T0L})$ *is closed under union.*

Proof. Let $G_i = (\Sigma_i, H_i, \omega_i, \Delta_i, k, noc_i)$, $i = 1, 2$, be a forbidding random context uklE(P) T0L systems. We construct a forbidding random context uklE(P)T0L system

$$G = (\Sigma, H, S, \Delta, k, noc)$$

with $L(G) = L(G_1) \cup L(G_2)$. First, we set

$$\Sigma_1' = \{x' \mid x \in \Sigma_1\}, \qquad \Sigma_2'' = \{x'' \mid x \in \Sigma_2\}.$$

Especially, let $\Delta_1' = \{x' \mid x \in \Delta_1\}$, $\Delta_2'' = \{x'' \mid x \in \Delta_2\}$.

For a word $w = a_1 \ldots a_n \in \Sigma_1^*$, let $w' = a_1' \ldots a_n' \in \Sigma_1'^*$. Analogously, $w'' \in \Sigma_2''^*$ is given. Then we define

$$\Sigma = \Sigma_1' \cup \Sigma_2'' \cup \Delta_1 \cup \Delta_2 \cup \{S\}, \qquad \Delta = \Delta_1 \cup \Delta_2, \qquad H = H_1' \cup H_2'' \cup \{h_I, h_T', h_T''\},$$

where $H_1' = \{h' \mid h \in H_1\}$, $H_2'' = \{h'' \mid h \in H_2\}$. For $h \in H_1$, the table $h' \in H_1'$ is determined by

$$h'(x') = \{w' \mid w \in h(x)\} \quad \text{for } x \in \Sigma_1, \qquad h'(y) = \{y\} \quad \text{for } y \in \Sigma \setminus \Sigma_1',$$
$$noc(h') = \{x' \mid x \in noc(h)\} \cup \Sigma_2'' \cup \Delta_1.$$

Analogously, by exchanging 1 with 2 and $'$ with $''$, $h'' \in H_2''$ can be constructed. The initial table h_I and the terminal table h_T' (the terminal table h_T'' is defined analogously) are given by

$$h_I(S) = \{\omega_1', \omega_2''\}, \qquad h_I(x) = \{x\} \quad \text{for } x \in \Sigma \setminus \{S\}, \qquad noc(h_I) = \Sigma \setminus \{S\}$$

and

$$h_T'(x') = \{x\} \quad \text{for } x \in \Delta_1, \qquad h_T'(x) = \{x\} \quad \text{for } x \in \Sigma \setminus \Delta_1',$$
$$noc(h_T') = \Sigma \setminus (\Delta_1' \cup \Delta_1).$$

Obviously, $L(G) = L(G_1) \cup L(G_2)$. □

By Theorem 2.3.4 of [3] we know that

$$\mathscr{L}(\mathrm{fRC}, \mathrm{cf}(\text{-}\varepsilon)) = \mathscr{L}(\mathrm{O}, \mathrm{cf}(\text{-}\varepsilon))$$

where $\mathscr{L}(\mathrm{O}, \mathrm{cf}(\text{-}\varepsilon))$ is the family of languages generated by ordered (ε-free) context-free grammars. In [6, Theorem 5.2], see also [7], it is proved that

$$\mathscr{L}(\mathrm{fRC}, \mathrm{cf}(\text{-}\varepsilon)) = \mathscr{L}(\mathrm{O}, \mathrm{cf}(\text{-}\varepsilon)) \subsetneq \mathscr{L}(\mathrm{P}, ut, \mathrm{cf}(\text{-}\varepsilon)). \tag{5}$$

By Theorems 5.2 and 3.1, it follows that

$$\mathscr{L}(frand, \mathrm{u1lE(P)T0L}) \subsetneq \mathscr{L}(\mathrm{P}, ac, \mathrm{cf}(\text{-}\varepsilon)).$$

For other limitations $k \in \mathbb{N}$, $k > 1$, we shall see that instead of the strict inclusion we get the equality. Altogether, we have

Theorem 5.5. *For all* $k, k' \in \mathbb{N}$, $k > 1$, $x \in \{\emptyset, p, ptv, m, gc, rc\}$,

$$\begin{aligned}\mathscr{L}(frand, \mathrm{u1lE(P)T0L}) &\subsetneq \mathscr{L}(x, frand, \mathrm{uklE(P)T0L}) \\ &= \mathscr{L}(x, rand\text{-}app, \mathrm{uk'lE(P)T0L}) = \mathscr{L}(\mathrm{P}, ac, \mathrm{cf}(\text{-}\varepsilon)).\end{aligned}$$

Proof. By the definitions and by [21] (Theorems 3.2–3.4 and 5.4), it is known that for all $k \in \mathbb{N}$, $x \in \{p, ptv, m, gc, rc\}$,

$$\begin{aligned}\mathscr{L}(frand, \mathrm{uklEPT0L}) \subseteq \mathscr{L}(x, frand, \mathrm{uklEPT0L}) &\subseteq \mathscr{L}(x, rand\text{-}app, \mathrm{uklEPT0L}) \\ = \mathscr{L}(rand\text{-}app, \mathrm{uklEPT0L}) &\subseteq \mathscr{L}(\mathrm{P}, ac, \mathrm{cf}\text{-}\varepsilon)\end{aligned}$$

(where in case $k = 1$ the last inclusion is already known to be an equality) and

$$\begin{aligned}\mathscr{L}(frand, \mathrm{uklET0L}) \subseteq \mathscr{L}(x, frand, \mathrm{uklET0L}) &\subseteq \mathscr{L}(x, rand\text{-}app, \mathrm{uklET0L}) \\ = \mathscr{L}(rand\text{-}app, \mathrm{uklET0L}) &= \mathscr{L}(\mathrm{P}, ac, \mathrm{cf})\end{aligned}$$

in the non-propagating case. We have to prove that in the case $k > 1$, $\mathscr{L}(\mathrm{P}, ac, \mathrm{cf}(\text{-}\varepsilon)) \subseteq \mathscr{L}(frand, \mathrm{uklE(P)T0L})$.

We carry out the proof for the propagating case. It is easily seen that it is also true in the non-propagating case.

Consider $L \in \mathscr{L}(\mathrm{P}, ac, \mathrm{cf}\text{-}\varepsilon)$, $L \subseteq \Delta^*$, and let $k \in \mathbb{N}$, $k > 1$. Since $\mathscr{L}(\mathrm{P}, ac, \mathrm{cf}\text{-}\varepsilon)$ is an AFL (see [3, Theorem 1.3.2]), it is also closed under left derivatives (see [3, Corollary of Theorem 1.3.2]). Thus, $L_w = \{v \mid wv \in L,\ v \in \Delta^*,\ |v| > 0\} \in \mathscr{L}(\mathrm{P}, ac, \mathrm{cf}\text{-}\varepsilon)$ for every $w \in \Delta^k$.

Now, L may be represented as $L = \bigcup_{w \in \Delta^k} \{w\} L_w \cup L'$, where L' is some finite ε-free language. We want to show that $L \in \mathscr{L}(frand, \mathrm{uklEPT0L})$. Obviously, every finite ε-free language is contained in $\mathscr{L}(frand, \mathrm{uklEPT0L})$. Because of the closure of $\mathscr{L}(frand, \mathrm{uklEPT0L})$ with respect to union shown in Lemma 5.4, it remains to prove that $\{w\} L_w \in \mathscr{L}(frand, \mathrm{uklEPT0L})$ for every $w \in \Delta^k$.

Let $w = a_1 \dots a_k \in \Delta^k$ be a fixed word. Let L_w be generated by some programmed ε-free context-free grammar $G_1 = (V_N, \Delta, S_1, P, \sigma, \mu)$ with appearance checking. We construct a forbidding random context uklEPT0L system $G = (\Sigma, \Delta, H, S, noc)$ with $L(G) = \{w\}L(G_1)$. Let

$$\Sigma = V_N \cup V_N' \cup \mathrm{Lab}(P) \cup \mathrm{Lab}(P)' \cup \{S, F, \$_1, \dots, \$_{k-1}, \#_1, \dots, \#_{k-2}\}$$

where

$$V_N' = \{x' \mid x \in V_N\}, \qquad \mathrm{Lab}(P)' = \{p' \mid p \in \mathrm{Lab}(P)\},$$

$$H = \{h_I, h_T\} \cup \{h_p^-, h_{p,1}^+, h_{p,2}^+ \mid p \in \mathrm{Lab}(P)\}.$$

Defining the tables we assume that if for some $x \in \Sigma$, $h \in H$, $h(x)$ is not written down explicitly or if we would get $h(x) = \emptyset$ according to our definition given below, then we mean $h(x) = \{F\}$. First we set

$$h_I(S) = \{\$_1 \dots \$_{k-1} p S_1 \mid p \in \mathrm{Lab}(P)\}, \qquad noc(h_I) = \emptyset,$$
$$h_T(\$_i) = \{a_i\}, \quad i = 1, \dots, k-1, \qquad h_T(p) = \{a_k\} \quad \text{for } p \in Lab(P),$$
$$noc(h_T) = \Sigma \setminus \Delta.$$

Furthermore, for every $p = (A \to v, \sigma(p), \mu(p)) \in P$ with success field $\sigma(p)$ and failure field $\mu(p)$, we get three tables h_p^-, $h_{p,1}^+$, $h_{p,2}^+$ defined as follows:

$$h_p^-(p) = \mu(p), \qquad h_p^-(\$_i) = \{\$_i\} \quad \text{for } i = 1, \dots, k-1,$$
$$noc(h_p^-) = \{A\} \cup (\mathrm{Lab}(P) \setminus \{p\}) \cup \mathrm{Lab}(P)' \cup V_N' \cup \{\#_1, \dots, \#_{k-2}\},$$

$$h_{p,1}^+(p) = \{p'\}, \qquad h_{p,1}^+(A) = \{A'\}, \qquad h_{p,1}^+(\$_i) = \{\#_i\} \quad \text{for } i = 1, \dots, k-2,$$
$$noc(h_{p,1}^+) = (\mathrm{Lab}(P) \setminus \{p\}) \cup \mathrm{Lab}(P)' \cup V_N' \cup \{\#_1, \dots, \#_{k-2}\},$$

$$h_{p,2}^+(A') = \{v\}, \qquad h_{p,2}^+(p') = \sigma(p), \qquad h_{p,2}^+(\#_i) = \{\$_i\} \quad \text{for } i = 1, \dots, k-2,$$
$$noc(h_{p,2}^+) = \mathrm{Lab}(P) \cup (\mathrm{Lab}(P)' \setminus \{p'\}) \cup \{\$_1, \dots, \$_{k-2}\}.$$

A derivation step $w_1 \Rightarrow_p w_2$, $w_1 = u_1 A u_2$, $w_2 = u_1 v u_2$, according to G_1 is simulated by the derivation

$$\$_1 \dots \$_{k-2}\$_{k-1} p w_1 \Rightarrow_{h_{p,1}^+} \#_1 \dots \#_{k-2}\$_{k-1} p' u_1 A' u_2 \Rightarrow_{h_{p,2}^+} \$_1 \dots \$_{k-2}\$_{k-1} q u_1 v u_2$$

with $q \in \sigma(p)$. h_p^- is not applicable (nor h_I or h_T). A derivation step $w_1 \Rightarrow_p w_2$ where A is not contained in w_1, is simulated by

$$\$_1 \dots \$_{k-1} p w_1 \Rightarrow_{h_p^-} \$_1 \dots \$_{k-1} q w_1$$

with $q \in \mu(p)$. The other tables introduce the failure symbol F or are not applicable. It follows that $L(G_1) = \{w\}L_w$. □

Comparing uniformly limited versus limited systems in this case, we obtain, using the results of this section together with Eq. (5) above:

Theorem 5.6. *For every $k,k' \in \mathbb{N}$, $k' > 1$, we have*

$$\mathscr{L}(\mathit{frand}, k\text{lE(P)T0L}) \subseteq \mathscr{L}(\mathit{frand}, \text{u}k'\text{lE(P)T0L}),$$

$$\mathscr{L}(\mathit{frand}, \text{1lE(P)T0L}) \supsetneq \mathscr{L}(\mathit{frand}, \text{u1lE(P)T0L}).$$

Proof. As regards the first relation, we know

$$\mathscr{L}(\mathit{frand}, k\text{lE(P)T0L}) \overset{\text{Th. 5.1}}{=} \mathscr{L}(k\text{lE(P)T0L}) \tag{6}$$

$$\overset{\text{[4, Th. 4.5, Th 4.6]}}{\subseteq} \mathscr{L}(\text{P}, ut, \text{cf}(\text{-}\varepsilon)) \tag{7}$$

$$\subseteq \mathscr{L}(\text{P}, ac, \text{cf}(\text{-}\varepsilon)) \tag{8}$$

$$\overset{\text{Th. 5.5}}{=} \mathscr{L}(\mathit{frand}, \text{u}k'\text{lE(P)T0L}). \tag{9}$$

We note that in [4, Theorems 4.5 and 4.6] it has been shown that the family of periodic function limited E(P)T0L languages is included in $\mathscr{L}(\text{P}, ut, \text{cf}(\text{-}\varepsilon))$. Obviously, the constant k can be considered as a periodic limitation function. As regards the second claim, we know that

$$\mathscr{L}(\mathit{frand}, \text{u1lE(P)T0L}) \overset{\text{Th. 5.2}}{=} \mathscr{L}(\text{fRC}, \text{cf}(\text{-}\varepsilon))$$

$$\overset{\text{Eq. (5)}}{\subsetneq} \mathscr{L}(\text{P}, ut, \text{cf}(\text{-}\varepsilon))$$

$$\overset{\text{Th. 4.1}}{=} \mathscr{L}(\text{1lE(P)T0L})$$

$$\subseteq \mathscr{L}(\mathit{frand}, \text{1lE(P)T0L}). \qquad \square$$

The strictness of the first relation above is not known, but closely related to the question whether unconditional transfer is less powerful than appearance checking in programmed grammars; see [9] and the inclusion chain in Eqs. (6)–(9). We pose the exact relation in the case $k > 1$, $k' = 1$ as an open problem. Let us discuss this question a bit further.

We have remarked in [9, Theorem 3.1] that $\mathscr{L}(k\text{lET0L})$ contains non-recursive languages. (The proof idea is to construct klET0L machines, an analogue of 1lET0L machines which have been treated in [6].) By [16, Theorem 4.14], $\mathscr{L}(k\text{lET0L})$ is closed under arbitrary homomorphisms. Obviously, for every $L \in \mathscr{L}(k\text{lET0L})$ there exists an erasing homomorphism h and a language $L' \in \mathscr{L}(k\text{lEPT0L})$ such that $L = h(L')$. Therefore, there exists a non-recursive language representable as homomorphic image of some $L \in \mathscr{L}(k\text{lE(P)T0L})$.

On the other hand, the class $\mathscr{L}(\text{O}, \text{cf})$ (which coincides with $\mathscr{L}(\mathit{frand}, \text{u1lET0L})$ by Eq. (5) and Theorem 5.2) contains only recursive languages [1, Corollary 3.8]

and is closed under arbitrary homomorphisms [3, Table 2.5.1]. Therefore, there does not exist any non-recursive language representable as homomorphic image of any $L \in \mathscr{L}(frand, \mathrm{u1lE(P)T0L})$.

This observation readily implies:

Corollary 5.1. *For every* $k \in \mathbb{N}$, $\mathscr{L}(frand, k\mathrm{lE(P)T0L}) \neq \mathscr{L}(frand, \mathrm{u1lE(P)T0L})$.

It would be interesting to know whether the inclusion

$$\mathscr{L}(\mathrm{O}, \mathrm{cf}(\text{-}\varepsilon)) \subseteq \mathscr{L}(frand, k\mathrm{lE(P)T0L})$$

is true for any $k > 1$.

6. Permitting random context

In this section, we consider the families of languages generated by (permitting) random context klE(P)T0L and uklE(P)T0L systems. We already know the results listed in Eq. (3). We shall establish some relations to the families of languages generated by (recurrent) programmed context-free grammars or random context ET0L systems. In [18, Theorem 5.3] it has been demonstrated that $\mathscr{L}(rand, \mathrm{ET0L}) \subsetneqq \mathscr{L}(rand, k\mathrm{lET0L})$ where the last family equals $\mathscr{L}(\mathrm{P}, ac, \mathrm{cf})$ (equal to $\mathscr{L}(\mathrm{re})$). First, we prove a similar result for propagating systems. We cannot carry over the corresponding proofs of [18], since they are based on constructions using erasing production. But the proof of our paper is also valid for the non-propagating case.

Theorem 6.1. *For all* $k \in \mathbb{N}$,

$$\mathscr{L}(rand, \mathrm{E(P)T0L}) \subsetneqq \mathscr{L}(rand, k\mathrm{lE(P)T0L}) \subsetneqq \mathscr{L}(\mathrm{P}, ac, \mathrm{cf}(\text{-}\varepsilon)).$$

Proof. Let $G = (\Sigma, H, S, \Delta, oc)$ be a random context EPT0L system where $H = \{h_1, \ldots, h_n\}$ for some $n \in \mathbb{N}$ and, without loss of generality, $S \in \Sigma - \Delta$. We define a random context klEPT0L system $G' = (\Sigma', H', S, \Delta, k, oc')$ as follows. Set

$$\Sigma' = \Sigma \cup \bigcup_{i=1}^{n} \{X_a^i \mid a \in \Sigma\} \cup \bigcup_{i=1}^{n} \{Y_a^i \mid a \in \Sigma\} \cup \{F\} \quad \text{and} \quad H' = \bigcup_{i=1}^{n} \{h_{i1}, h'_{i,1}, h_{i2}, h_{i3}\}$$

with

$$\begin{aligned}
&h_{i1}(a) = \{Y_a^i\}, \quad && h_{i1}(Y_a^i) = \{F\}, \quad && h_{i1}(X_a^i) = \{F\},\\
&h'_{i1}(a) = \{Y_a^i\}, \quad && h'_{i1}(Y_a^i) = \{Y_a^i\}, \quad && h'_{i1}(X_a^i) = \{F\},\\
&h_{i2}(a) = \{F\}, \quad && h_{i2}(Y_a^i) = \{X_a^i\}, \quad && h_{i2}(X_a^i) = \{X_a^i\},\\
&h_{i3}(a) = \{a\}, \quad && h_{i3}(Y_a^i) = \{F\}, \quad && h_{i3}(X_a^i) = h_i(a)
\end{aligned}$$

for $i = 1, \ldots, n$ and $a \in \Sigma$. Furthermore, let $oc'(h_{i1}) = oc(h_i)$, $oc'(h'_{i1}) = \{Y_a^i \mid a \in oc(h_i)\}$ and $oc'(h_{i2}) = oc'(h_{i3}) = \emptyset$ for $i = 1, \ldots, n$. Observe that h_{i1} and h'_{i1} contain nearly the

same productions, but their occurrence sets are different. As usual, if for some $h \in H'$, $x' \in \Sigma'$, $h(x')$ is not explicitly written down, then $h(x') = \{F\}$. The construction is quite similar to that for non-random context systems in the proof of Theorem 4.3 in [16] so that we do not go into the details of the equivalence proof. We only note that in addition, we had to take care of the occurrence check which has been managed by substituting every table h_{i1} of the construction of [16] by two tables h_{i1} and h'_{i1}. By one application of h_{i1} followed by possibly several applications of h'_{i1}, all symbols $a \in oc'(h_{i1})$ are replaced by symbols $Y_a^i \in oc'(h'_{i1})$.

The second inclusion is proved by a construction being a mixture of the constructions in the proof of Theorem 1 in [2] and Theorem 4.1 in [22]. Furthermore, we have to take into account the random context by some checking productions. Let $G = (\Sigma, H, \omega, \Delta, k, oc)$ be a random context klEPT0L system. We construct an ε-free context-free programmed grammar as follows. Let

$$\Sigma = \{a_1, \ldots, a_n\},\ \Delta = \{a_1, \ldots, a_r\},\ H = \{h_1, \ldots, h_m\},\ oc(h_j) = \{a_{j1}, \ldots, a_{jq_j}\}$$

for some $n, m, r \in \mathbb{N}$ with $r \leqslant n$ and $q_j \in \mathbb{N}_0$ for $j = 1, \ldots, m$. We assume that $oc(h_j) \neq \emptyset$ for $j = 1, \ldots, m'$ and $oc(h_j) = \emptyset$ for $j = m' + 1, \ldots, m$ for some $m' \in \mathbb{N}_0$, $0 \leqslant m' \leqslant m$. Let $\Sigma' = \{A_1, \ldots, A_n\}$ be a new alphabet, and $g : \Sigma'^* \to \Sigma^*$ the bijective homomorphism defined by $g(A_i) = a_i$. Now we define the ε-free context-free programmed grammar $G' = (V_N, V_T, X_0, P, \sigma, \mu)$ where

$$V_N = \{A_i \mid 1 \leqslant i \leqslant n\} \cup \{A_{i,j} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m\} \cup \{X_0\} \quad \text{and} \quad V_T = \Delta.$$

The set of labels of P is given by

$$\begin{aligned} \mathrm{Lab}(P) = {} & \{f_0, f_1, \ldots, f_r\} \cup \{t_{jv_j} \mid 1 \leqslant j \leqslant m', 1 \leqslant v_j \leqslant q_j\} \\ & \cup \{f_{i,j,\kappa} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m, 1 \leqslant \kappa \leqslant k\} \cup \bigcup_{i=1}^{n} \bigcup_{j=1}^{m} P_{i,j} \end{aligned}$$

where the sets of labels $P_{i,j}$ are implicitly given below. We define

$$\begin{aligned} f_0 &= (X_0 \to g^{-1}(\omega), \{f_1, \ldots, f_r\} \cup \{t_{j1} \mid 1 \leqslant j \leqslant m'\} \\ &\quad \cup \{f_{1j1} \mid m' + 1 \leqslant j \leqslant m\}, \emptyset), \\ f_\rho &= (A_\rho \to a_\rho, \{f_1, \ldots, f_r\}, \emptyset) \quad \text{for } 1 \leqslant \rho \leqslant r, \\ t_{jv_j} &= (A_{v_j} \to A_{v_j}, \{t_{j(v_j+1)}\}, \emptyset) \quad \text{for } 1 \leqslant j \leqslant m', 1 \leqslant v_j \leqslant q_j - 1, \\ t_{jq_j} &= (A_{q_k} \to A_{q_k}, \{f_{1j1}\}, \emptyset) \quad \text{for } 1 \leqslant j \leqslant m', \\ f_{ij\kappa} &= (A_i \to A_{ij}, \{f_{ij(\kappa+1)}\}, \{f_{(i+1)j1}\}) \\ &\qquad \text{for } 1 \leqslant i \leqslant n-1,\ 1 \leqslant j \leqslant m, 1 \leqslant \kappa \leqslant k-1, \\ f_{ijk} &= (A_i \to A_{ij}, \{f_{(i+1)j1}\}, \{f_{(i+1)j1}\}) \quad \text{for } 1 \leqslant i \leqslant n-1, 1 \leqslant j \leqslant m, \end{aligned}$$

$$f_{njk} = (A_n \to A_{nj}, \{f_{nj(k+1)}\}, P_{1j}) \quad \text{for } 1 \leqslant j \leqslant m, 1 \leqslant k \leqslant k-1,$$
$$f_{njk} = (A_n \to A_{nj}, P_{1j}, P_{1j}) \quad \text{for } 1 \leqslant k \leqslant k-1,$$
$$P_{ij} = \{(A_{ij} \to g^{-1}(v), P_{ij}, P_{(i+1)j}) \mid v \in h_j(a_i)\} \quad \text{for } 1 \leqslant i \leqslant n-1, 1 \leqslant j \leqslant m,$$
$$P_{nj} = \{(A_{nj} \to g^{-1}(v), P_{nj}, \{f_1, \ldots, f_r\} \cup \{t_{j'1} \mid 1 \leqslant j' \leqslant m'\}$$
$$\cup \leqslant \{f_{1j'1} \mid m'+1 \leqslant j' \leqslant m\})\} \quad \text{for } 1 \leqslant j \leqslant m.$$

The proof of $L(G) = L(G')$ is similar to those in [2, 22] and is omitted. □

It is a natural question to ask which of the inclusion relations given above are strict and which are not. Unfortunately, very little is known here, especially in case of propagating systems. From [21, Theorem 5.2], we know that $\mathscr{L}(rand, k\text{IET0L}) = \mathscr{L}(\text{re})$ for each $k \in \mathbb{N}$. We can only state the following analogous theorem for the case $k = 1$. As remarked above, this situation is similar to the case without random context conditions, see Theorem 4.1.

Lemma 6.2. *For every $k \in \mathbb{N}$, $\mathscr{L}(rand, k\text{IE(P)T0L})$ is closed under union.*

Proof. The proof is nearly the same as that of Lemma 5.4. But defining h', we have instead of the non-occurrence set $oc(h') = \{x' \mid x \in oc(h)\}$ and furthermore, we must set $h'(x) = \{F\}$ for $x \in \Sigma_2'' \cup \Delta_1 \cup \{F\}$ where F is a new failure symbol. In a similar manner, the other tables have to be changed. □

Theorem 6.3. *We have $\mathscr{L}(rand, 1\text{IE(P)T0L}) = \mathscr{L}(\text{P}, ac, \text{cf}(\text{-}\varepsilon))$.*

Proof. According to Theorem 6.1, it remains to prove $\mathscr{L}(\text{P}, ac, \text{cf}(\text{-}\varepsilon)) \subseteq \mathscr{L}(rand, 1\text{IE(P)T0L})$. Let $L \in \mathscr{L}(\text{P}, ac, \text{cf}(\text{-}\varepsilon))$, $L \subseteq \Delta^*$. Since $\mathscr{L}(\text{P}, ac, \text{cf}(\text{-}\varepsilon))$ is a trio, $L_a = \{w \in \Delta^+ \mid aw \in L\} \in \mathscr{L}(\text{P}, ac, \text{cf}(\text{-}\varepsilon))$ for every $a \in \Delta$. Let $G = (V_N, \Delta, P, S, \sigma, \mu)$ be a (propagating) context-free programmed grammar with appearance checking generating L_a. We construct a 1IE(P)T0L system $G' = (\Sigma, \Delta, H, S', 1, oc)$ with permitting context generating $\{a\}L_a$. Let $\Sigma = V_N \cup \text{Lab}(P) \cup \{S', F\} \cup \Delta$. H contains the following productions: (As usual, "incomplete tables" can be supplemented by productions of the form $X \to F$.)

1. one initialization table $h_{init}(S') = \text{Lab}(P)\{S\}$;
2. one termination table $h_{term}(b) = \{b\}$ for $b \in \Delta$, $h_{term}(p) = \{a\}$ for $p \in \text{Lab}(P)$;
3. for every rule $(p : A \to w, \sigma(p), \mu(p))$, we have two tables, namely
   (a) h_p^+ with $h_p^+(A) = \{w\}$, $h_p^+(p) = \sigma(p)$, and $h_p^+(B) = \{B\}$ for every $B \in (V_N \setminus \{A\}) \cup \Delta$, $oc(h_p^+) = \{A, p\}$; and
   (b) h_p^- with $h_p^-(p) = \mu(p)$, and $h_p^-(B) = \{B\}$ for every $B \in (V_N \setminus \{A\}) \cup \Delta$, $oc(h_p^-) = \{p\}$. □

We turn to uniformly limited systems in the following.

Theorem 6.4. *For every $k \in \mathbb{N}$, we have* $\mathscr{L}(rand, \mathrm{uk1E(P)T0L}) \subsetneqq \mathscr{L}(\mathrm{P}, \mathrm{cf}(\text{-}\varepsilon))$.

Proof. It is easy to adapt a proof outline showing $\mathscr{L}(\mathrm{uk1E(P)T0L}) \subsetneqq \mathscr{L}(\mathrm{P}, \mathrm{cf}(\text{-}\varepsilon))$, as sketched in[10, p. 57f].

Let $L \in \mathscr{L}(rand, \mathrm{uk1E(P)T0L})$ be generated by the system $G = (\Sigma, H, \omega, \Delta, k, oc)$. Let ℓ be the length of the longest right-hand side of some production in G. We define the simulating grammar $G' = (V_N, V_T = \Delta, S, P, \sigma)$ with $V_N = \{A', A'' \mid A \in \Sigma\} \cup \{S\}$ (furthermore, we interpret $'$ and $''$ as homomorphisms) in the following:

$$\begin{aligned}\mathrm{Lab}(P) = {} & \{[v] \mid \omega \overset{*}{\Rightarrow}_G v,\ |v| \leqslant k\ell\} \cup \{[\omega]\} \\ & \cup \{[h,i] \mid h \in H,\ oc(h) = \{A_1, \ldots, A_n\} \neq \emptyset,\ 1 \leqslant i \leqslant n\} \\ & \cup \{[h,i,A] \mid h \in H,\ 0 \leqslant i \leqslant k-1,\ A \in \Sigma\} \\ & \cup \{[h,i,B,w] \mid h \in H,\ 0 \leqslant i \leqslant k-1,\ w \in h(B),\ B \in \Sigma\} \cup \Delta.\end{aligned}$$

Moreover, let $\mathrm{sim} = \mathrm{sim}_1 \cup \mathrm{sim}_2$ denote the start labels of a simulation phase of some table, where

$$\begin{aligned}\mathrm{sim}_1 &= \{[h,i] \mid h \in H,\ oc(h) = \{A_1, \ldots, A_n\} \neq \emptyset,\ 1 \leqslant i \leqslant n\}, \\ \mathrm{sim}_2 &= \{[h,0,A] \mid h \in H,\ A \in \Sigma,\ oc(h) = \emptyset\}.\end{aligned}$$

P contains the following productions:

1. For every sentential form $v \in \Sigma^*$ which is generable by G and which is of length $l \leqslant k\ell$, we add a production $([v] : S \to v', \mathrm{sim} \cup \Delta, \emptyset)$. Moreover, we add $([\omega] : S \to \omega', \mathrm{sim} \cup \Delta, \emptyset)$.

(Admittedly, at this stage, it is not clear that collecting these v can be done algorithmically if we allow erasing productions. Nevertheless, since we give an equivalent programmed grammar without appearance checks, and since $\mathscr{L}(\mathrm{P}, \mathrm{cf})$ is contained in the family of recursive languages, this problem is circumvented indirectly.)

2. One application of some table h with $oc(h) = \{A_1, \ldots, A_n\}$ is simulated by
   (a) (if $oc(h) \neq \emptyset$) a series of n productions of the form $([h,i] : A_i' \to A_i', \{[h, i+1]\}, \emptyset)$ (if $i < n$), and $([h,n] : A_n' \to A_n', \{[h,0,A] \mid h \in H,\ A \in \Sigma\}, \emptyset)$;
   (b) a series of k productions of the form $([h,i,A] : A' \to A'', H_i, \emptyset)$, where $H_i = \{[h, i+1, B] \mid B \in \Sigma\}$ if $i < k-1$, and $H_{k-1} = \{[h,0,B,w] \mid w \in h(B), B \in \Sigma\}$;
   (c) a series of k productions of the form $([h,i,B,w] : B'' \to w', H_i', \emptyset)$, where $H_i' = \{[h, i+1, B, w] \mid w \in h(B),\ B \in \Sigma\}$ if $i < k-1$, and $H_{k-1}' = \mathrm{sim} \cup \Delta$.
3. For every $a \in \Delta$, there exists a terminating production $(a : a' \to a, \Delta, \emptyset)$.

Let $\omega = w_0 \Rightarrow_G w_1 \Rightarrow_G \cdots \Rightarrow_G w_I = w$ be a derivation of w according to G. Let $0 \leqslant i \leqslant I$ be the greatest index such that $|w_i| \leqslant k\ell$ (with $i = 0$ if no such index exists). Then, we have $S \Rightarrow_{G'} w_i'$, and the derivation sequence $w_i \Rightarrow_{G,h_i} w_{i+1} \Rightarrow_{G,h_{i+1}} \cdots \Rightarrow_{G,h_{I-1}} w_I = w$ can be simulated by $w_i' \Rightarrow_{G'}^{|oc(h_i)|+2k} w_{i+1}' \Rightarrow_{G'}^{|oc(h_{i+1})|+2k} \cdots \Rightarrow_{G'}^{|oc(h_{I-1})|+2k} w_I' = w'$.

If $w \in \Delta^*$, we can now obtain w from w' applying productions labelled with symbols from Δ $|w|$ times.

On the other hand, every successful derivation in G' has one the following two forms:

1. $S \Rightarrow_{G'} w_0' = w' \Rightarrow_{G'}^{|w'|} w \in \Delta^*$ in case $|w_0| \leqslant \ell k$, $w_0 \in \Delta^*$ By our definition, $w_0 \in L(G)$.

2. $S \Rightarrow_{G'} w_0' \Rightarrow_{G'}^{|oc(h_1)|+2k} w_1' \Rightarrow_{G'}^{|oc(h_2)|+2k} \cdots \Rightarrow_{G'}^{|oc(h_{l-1})|+2k} w_l' = w' \Rightarrow_{G'}^{|w'|} w \in \Delta^*$. Now, every w_i' has length $l \geqslant k$. Therefore, to such a derivation, there corresponds a derivation $\omega \overset{*}{\Rightarrow}_G w_0 \Rightarrow_{G,h_1} w_1 \Rightarrow_{G,h_2} \cdots \Rightarrow_{G,h_{l-1}} w_l = w \in \Delta^*$ in the original uniformly k-limited ET0L system G. □

Interestingly, there are nice connections with recurrent programmed grammar.

Theorem 6.5. *For every $k > 1$, we have $\mathscr{L}(\mathrm{RP}, \mathrm{cf}(\text{-}\varepsilon)) \subsetneqq \mathscr{L}(rand, \mathrm{u}k\mathrm{lE(P)T0L})$.*

Proof. First, we deal with the case admitting erasing productions. Let $L \in \mathscr{L}(\mathrm{RP}, \mathrm{cf})$, $L \subsetneqq \Delta^*$. Let $G = (V_N, \Delta, S, P, \sigma)$ be a recurrent programmed grammar generating L. We construct a uklET0L system $G' = (\Sigma, H, S', \Delta, k, oc)$ with permitting context generating L. Let $\Sigma = V_N \cup \mathrm{Lab}(P) \cup \mathrm{Lab}(P)' \cup (V_N \times \mathrm{Lab}(P)) \cup \{S', F\}$. H contains the following productions: (As usual, "incomplete tables" can be supplemented by productions of the form $X \to F$.)

1. one initialization table $h_{init}(S') = \mathrm{Lab}(P)\{S\}$;
2. one termination table $h_{term}(b) = \{b\}$ for $b \in \Delta$, $h_{term}(p) = \{\varepsilon\}$ for $p \in \mathrm{Lab}(P)$;
3. for every rule $(p : A \to w, \sigma(p))$, we have two tables, namely
   (a) $h_{p,1}(A) = \{(A, p), A\}$, $h_{p,1}(p) = \{p'\}$, $oc(h_{p,1}) = \{A, p\}$, and
   (b) $h_{p,2}(p') = \sigma(p)$, $h_{p,2}((A, p)) = \{w\}$, $oc(h_{p,2}) = \{p', (A, p)\}$.

Observe that every rule-simulating table needs a special marker p or p' in order to be applicable. Especially, this prevents the grammar from erasing the marker using the termination table prematurely, erroneously continuing the derivation process afterwards. If $h_{p,1}$ is applied without changing p to p', it may be applied a certain number of times, since G is recurrent.

Observing the following facts, it is easy work to adapt the proof above for the propagating case.

1. $\mathscr{L}(\mathrm{RP}, \mathrm{cf}\text{-}\varepsilon)$ is a trio according to Theorem 3.4. Especially, it is closed under derivatives.

2. $\mathscr{L}(\mathrm{u}k\mathrm{lEPT0L})$ is closed under union and contains the finite languages. (The proof of [22, Theorem 5.1] works also in the propagating case.)

Therefore, if $L \in \mathscr{L}(\mathrm{RP}, \mathrm{cf}\text{-}\varepsilon)$, $L \subsetneqq \Delta^*$, then $L_a := \{w \in L \mid w \in \Delta^+, aw \in L\} \in \mathscr{L}(\mathrm{RP}, \mathrm{cf}\text{-}\varepsilon)$. We take the same construction as above transforming a recurrent programmed grammar G_a generating L_a into a uniformly limited system generating $\{a\}L_a$, only changing the terminating table into $h_{term}(b) = \{b\}$ for $b \in \Delta$, $h_{term}(p) = \{a\}$ for $p \in \mathrm{Lab}(P)$. Hence, $L = \bigcup_{a \in \Delta} \{a\} L_a \cup L' \in \mathscr{L}(\mathrm{u}k\mathrm{lEPT0L})$, where L' is some finite language. □

We summarize our last results together with those obtained in Section 3 and Theorem 5.4 in [21]. The references are incorporated in the corresponding claims.

Theorem 6.6. *For every $k > 1$, we have*

$$\mathscr{L}(\mathrm{RC}, \mathrm{cf}(\text{-}\varepsilon)) \overset{\text{Th. 3.1}}{=} \mathscr{L}(rand, \mathrm{cf}(\text{-}\varepsilon)) \overset{[21,\ \text{Th. 5.2}]}{=} \mathscr{L}(rand, \mathrm{u1lE(P)T0L})$$

$$\overset{\text{Th. 3.2}}{\subseteq} \mathscr{L}(\mathrm{RP}, \mathrm{cf}(\text{-}\varepsilon))$$

$$\overset{\text{Th. 6.5}}{\subseteq} \mathscr{L}(rand, \mathrm{uk lE(P)T0L})$$

$$\overset{\text{Th. 6.4}}{\subseteq} \mathscr{L}(\mathrm{P}, \mathrm{cf}(\text{-}\varepsilon)).$$

Thus, we have found several classes "between" $\mathscr{L}(\mathrm{RC}, \mathrm{cf}(\text{-}\varepsilon))$ and $\mathscr{L}(\mathrm{P}, \mathrm{cf}(\text{-}\varepsilon))$, where the exact status of these inclusions is open. Nevertheless, these connections may give way to a solution of that old question.

Since appearance checking enhances the power of programmed grammars (see [12] or [13] together with [5] or [7]), we obtain when we consider uniform limited versus limited systems:

Corollary 6.1. *For every $k \in \mathbb{N}$,*

$$\mathscr{L}(rand, \mathrm{uk lE(P)T0L}) \overset{\text{Th. 6.4}}{\subseteq} \mathscr{L}(\mathrm{P}, \mathrm{cf}(\text{-}\varepsilon))$$

$$\subsetneq \mathscr{L}(\mathrm{P}, ac, \mathrm{cf}(\text{-}\varepsilon)) \overset{\text{Th. 6.3}}{=} \mathscr{L}(rand, \mathrm{1lE(P)T0L}).$$

Unfortunately, we do not know whether the relation $\mathscr{L}(rand, \mathrm{uk lE(P)T0L}) \subseteq \mathscr{L}(rand, k\mathrm{lE(P)T0L})$ holds true for every $k \in \mathbb{N}$. We can only show that the two language families do not coincide. More precisely, we get:

Corollary 6.2. *For every $k, k' \in \mathbb{N}$, $\mathscr{L}(rand, k\mathrm{lE(P)T0L}) \neq \mathscr{L}(rand, \mathrm{uk' lE(P)T0L})$.*

Proof. By Eq. (2), $L = \{a^{2^n} \mid n \in \mathbb{N}_0\} \in \mathscr{L}(k\mathrm{lE(P)T0L}) \subseteq \mathscr{L}(rand, k\mathrm{lE(P)T0L})$ for every $k \in \mathbb{N}$, but on the other hand (see proof of Theorem 3.3), we have $L \notin \mathscr{L}(\mathrm{P}, \mathrm{cf}(\text{-}\varepsilon))$. By Theorem 6.4, $L \notin \mathscr{L}(rand, \mathrm{uk' lE(P)T0L})$. □

References

[1] H. Bordihn and H. Fernau, Accepting grammars with regulation, *Internat. J. Comput. Math.* **53** (1994) 1–18.

[2] J. Dassow, A remark on limited 0L systems, *J. Inform. Processing Cybernet.* **EIK-24** (1988) 287–291.

[3] J. Dassow and Gh. Păun, *Regulated Rewriting in Formal Language Theory* (Akademie-Verlag, Berlin, 1989 & Springer, Berlin, 1990).

[4] H. Fernau, On function-limited Lindenmayer systems, *J. Inform. Processing Cybernet.* **EIK-27** (1991) 21–53.

[5] H. Fernau, Observations on grammar and language families, Tech. Report 22/94, Universität Karlsruhe (Germany), Fakultät für Informatik, August 1994. (Most of this report will appear in *Fund. Inform.*)

[6] H. Fernau, Membership for 1-limited ET0L languages is not decidable, *J. Inform. Processing Cybernet.* **EIK-30** (1994) 191–211.

[7] H. Fernau, A predicate for separating language classes, *EATCS Bull.* **56** (1995) 96–97.
[8] H. Fernau, A note on uniformly limited ET0L systems with unique interpretation, *Inform. Process. Lett.* **54** (1995) 199–204.
[9] H. Fernau, On unconditional transfer in: *Proc. MFCS'96.* Lecture Notes in Computer Science, Vol. 1113 (Springer, Berlin, 1996) 348–359.
[10] H. Fernau and H. Bordihn, Remarks on accepting parallel systems, *Internat. J. Comput. Math.* **56** (1995) 51–67.
[11] S. Ginsburg, *Algebraic and Automata-Theoretic Properties of Formal Languages* (North-Holland, Amsterdam, 1975).
[12] D. Hauschildt and M. Jantzen, Petri net algorithms in the theory of matrix grammars, *Acta Inform.* **31** (1994) 719–728.
[13] F. Hinz and J. Dassow, An undecidability result for regular languages and its applications to regulated rewriting, *EATCS Bull.* **38** (1989) 168–173.
[14] A. Salomaa, *Formal Languages* (Academic Press, New York, 1973).
[15] S.H. von Solms, Some notes on ET0L languages, *Internat. J. Comput. Math.* **5** (1976) 285–296.
[16] D. Wätjen, k-limited 0L systems and languages, *J. Inform. Processing Cybernet.* **EIK-24** (1988) 267–285.
[17] D. Wätjen, On k-uniformly-limited T0L systems and languages, *J. Inform. Processing Cybernet.* **EIK-26** (1990) 229–238.
[18] D. Wätjen, Regulation of k-limited ET0L systems, *Internat. J. Comput. Math.* **47** (1993) 29–41.
[19] D. Wätjen, Regulation of uniformly k-limited T0L systems, *J. Inform. Processing Cybernet.* **EIK-30** (1994) 169–187.
[20] D. Wätjen, On regularly controlled k-limited T0L systems, *Internat. J. Comput. Math.* **55** (1995) 57–66.
[21] D. Wätjen, Regulations of uniformly k-limited ET0L systems and their relations to controlled context-free grammars, *J. Automata Languages Combin.* **1** (1996) 55–74.
[22] D. Wätjen and E. Unruh, On extended k-uniformly-limited T0L systems and languages. *J. Inform. Processing Cybernet.* **EIK-26** (1990) 283–299.